

## Aplikasi Peningkatan Kualitas Citra Menggunakan Metode *Median Filtering* Untuk Menghilangkan *Noise*

Sadly Syamsuddin

Program Studi Teknik Informatika, STMIK Dipanegara, Makassar

e-mail: [sadlyg2@gmail.com](mailto:sadlyg2@gmail.com)

---

### Abstrak

Dilihat dari karakteristik dan bentuknya *noise* pada citra dibagi menjadi beberapa macam. *Noise* yang ada pada citra dianggap sangat mengganggu karena dapat mengurangi kualitas citra pada saat pencetakan, menyulitkan identifikasi pelaku kejahatan pada citra *cctv*, juga menyulitkan pada deteksi penyakit atau sel kanker pada citra kedokteran (*MRI*, *CTScan*, *XRay*). Untuk itu penulis merasa bahwa diperlukan suatu aplikasi yang dapat digunakan untuk mengurangi *noise* pada citra-citra yang mengalami gangguan tersebut. Metode yang digunakan dalam proses pengurangan citra dalam penelitian ini adalah teknik *median filtering* dimana metode ini memiliki beberapa langkah utama antara lain mencari nilai tengah dari pixel pixel pada window yang dievaluasi dan berulang seterusnya hingga citra menjadi bersih. Hasil dari penelitian menunjukkan presentasi pengurangan *noise* dapat dikatakan diatas 80%.

**Kata kunci** : citra , *noise* , *median filtering*

### Abstract

Judging from the characteristics and shape of the *noise* in the image is divided into several types. *Noise* in the image is considered very disturbing because it can reduce image quality during printing, making it difficult to identify criminals on *CCTV* images, also making it difficult to detect disease or cancer cells in medical images (*MRI*, *CTScan*, *XRay*). For this reason, the author feels that an application is needed that can be used to reduce *noise* in images that experience the disorder. The method used in the image extraction process in this study is a *median filtering* technique wherein this method has several main steps including looking for the middle value of pixel pixels in the evaluated and repeated windows until the image is clean. The results of the study show that the presentation of *noise* reduction can be said to be above 80%.

**Keywords**: image, *noise*, *median filtering*

### 1. PENDAHULUAN

Suatu data atau informasi disajikan tidak hanya berupa data teks tetapi juga dapat berupa audio, video, dan gambar. Pada zaman sekarang informasi sangatlah penting dan diperlukan, begitu juga informasi yang terdapat pada citra. Citra (*image*) atau istilah lain untuk gambar merupakan salah satu komponen multimedia yang berperan penting sebagai bentuk informasi visual dalam berbagai aspek kehidupan.

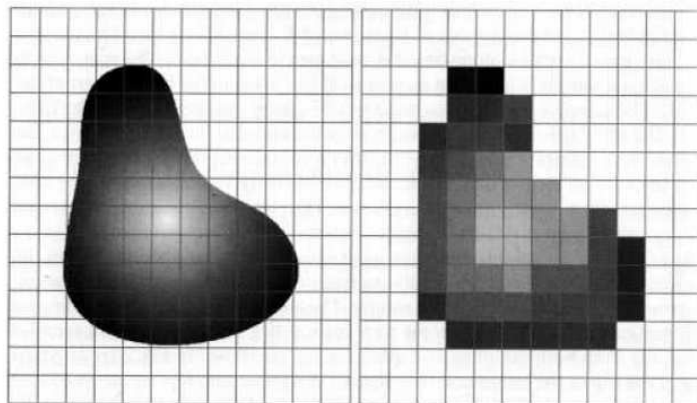
Dibandingkan dengan data teks citra memiliki banyak informasi dari pada data teks. Namun terkadang citra juga dapat mengalami penurunan yaitu degradasi atau penurunan mutu yang disebabkan oleh *derau* / *noise*, warna terlalu kontras, kabur, dan lain lain. Gangguan pada citra digital dapat terjadi karena banyak faktor, seperti kurangnya pencahayaan saat pengambilan gambar, keterbatasan resolusi pixel dari kamera yang digunakan, keterbatasan kemampuan menangkap gambar bergerak oleh kamera pengawas atau *cctv* (*closed circuit television*) yang disebabkan oleh terbatasnya memori dan buffer, interferensi gelombang elektro magnetik pada peralatan pencitraan kedokteran, dan sebagainya. Gambar atau pixel yang mengurangi kualitas citra dalam pengolahan citra disebut *derau* (*noise*). Dilihat dari karakteristik dan bentuknya *noise* pada citra dibagi menjadi beberapa macam yaitu *gaussian noise*, *speckle noise*, *salt and pepper noise*, *poisson noise*. *noise* pada citra dianggap sangat mengganggu karena dapat mengurangi kualitas citra pada saat pencetakan, menyulitkan identifikasi pelaku kejahatan pada citra *cctv*, juga menyulitkan pada deteksi penyakit atau sel kanker pada citra kedokteran (*MRI*, *CTScan*, *XRay*). Untuk itu penulis merasa bahwa diperlukan suatu aplikasi yang dapat mengurangi *noise* pada citra.

Dari pembahasan di atas maka jurnal ini akan membahas mengenai penelitian yang berjudul “Aplikasi Peningkatan Kualitas Citra Menggunakan Metode *Median Filtering* untuk Menghilangkan *Salt and Pepper Noise*”.

## 2. TINJAUAN PUSTAKA

### 2.1 Citra Digital

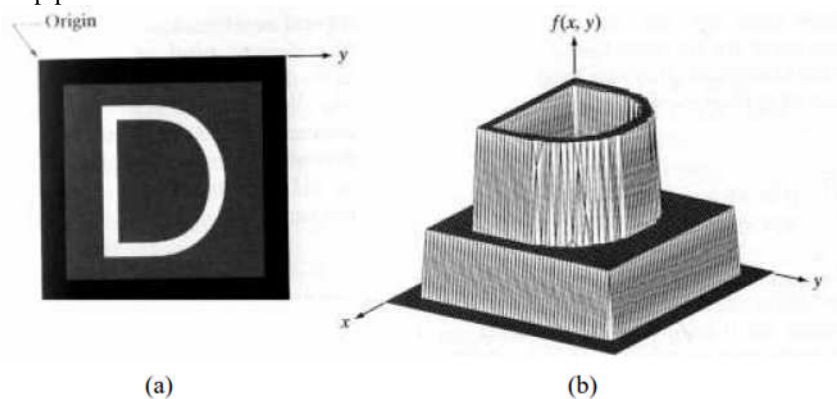
Suatu citra (gambar) analog dengan ukuran panjang kali lebar, dapat didigitalisasi dengan mengambil sampling berupa matriks berukuran  $m$  kali  $n$ , dengan  $m$  adalah jumlah sampling untuk panjang, dan  $n$  adalah jumlah sampling untuk lebar. Setiap sampling adalah berukuran bujur sangkar kecil. Semakin kecil ukuran sampling tersebut, semakin baik representasi citra kedalam bentuk digital, dan semakin halus tepian (edge) gambar yang dihasilkan. Gambar 1 dibawah ini menunjukkan contoh suatu gambar yang didigitalisasi dengan ukuran sampling yang masih cukup besar, sehingga tepian gambar akan berbentuk kasar (kotak-kotak).<sup>[2]</sup>



Gambar 1. Citra kontinu (analog) disampling oleh array sensor kuantisasi<sup>[2]</sup>

Sampling tersebut (selanjutnya disebut piksel, atau dalam bahasa Inggris adalah picture element = pictel menjadi pixel) yang terkecil dinyatakan dengan dot (atau berbentuk titik) karena ukuran yang sangat kecil. Matriks dot yang dihasilkan menyatakan derajat keabuan dari nilai sampling tersebut, untuk citra 8

bit bernilai 0 sampai 255. Untuk lebih jelas, citra digital seperti gambar 2.2 yang berbentuk suatu derajat keabuan (gray scale) dapat diplot pada grafik untuk memperlihatkan di level mana derajat keabuannya berada pada setiap piksel.



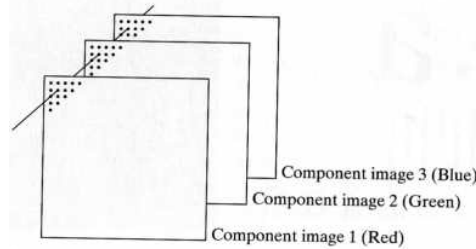
Gambar 2. Citra asli (a) Citra diplot pada suatu permukaan (b)<sup>[3]</sup>

$F(x,y)$  merupakan representasi dari nilai piksel pada setiap titik koordinat  $(x,y)$ , dengan titik origin dari matriks citra adalah  $(0, 0)$  di mana  $x$  dan  $y$  masing-masing adalah baris dan kolom. Untuk memudahkan penulisan matematis, maka matriks citra tersebut dapat dituliskan dalam bentuk persamaan (2.1) berikut ini.<sup>[1]</sup>

$$F(x,y) = A_{x,y} = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix} \dots\dots\dots [1]$$

Bila citra adalah citra berwarna, maka warna direpresentasikan oleh kanal kanal RGB (red-green-blue) seperti gambar 2.3, dengan setiap kanal dinyatakan dengan derajat keabuan dengan rentang 0 - 255. Misalnya untuk red, 0 menyatakan paling merah (pekat) dan 255 menyatakan paling pudar (putih). Demikian juga untuk kanal green (hijau) dan blue (biru).<sup>[4]</sup>

Sehingga matriks citra akan berdimensi ( m x n x 0 ) dengan m dan n adalah dimensi panjang dan lebar atau dimensi baris dan kolom pada matriks, dan adalah dimensi untuk kanal warna, yaitu 1 untuk kanal RED (merah), 2 untuk GREEN (hijau) dan 3 untuk BLUE (biru).<sup>[4]</sup>

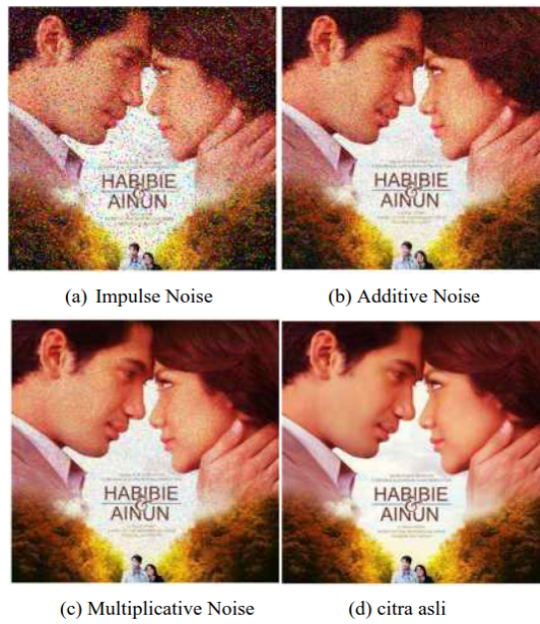


Gambar 3. Ilustrasi representasi kanal warna RGB pada citra<sup>[1]</sup>

**2.2 Noise Citra**

Noise (derau) merupakan penyebab utama penurunan kualitas citra (degradasi), sehingga sangat mengganggu apabila suatu foto ingin dicetak dan disimpan pada album atau dipajang diruang tamu. Dalam beberapa aplikasi medis, citra hasil pemindai MRI, CT Scan maupun USG, juga dapat terkena noise. Terlebih lagi USG, citra hitam putih analog yang dihasilkan sangat banyak memiliki noise, sehingga bila dibutuhkan analisa berbantuan komputer, perlu dilakukan preproses untuk memperbaiki kualitas citra (enhancement), agar deteksi selanjutnya bisa lebih tepat atau presisi. Ada 3 tipe noise yang umum pada pemrosesan citra digital, yaitu impulse noise, additive noise dan multiplicative noise. Ketiga tipe ini dapat dijelaskan sebagai berikut :<sup>[3]</sup>

1. *Impulse noise*, merupakan *noise* yang berbentuk sinyal impuls acak dan terdistribusi secara acak pula pada suatu citra digital. Adanya sinyal impuls ini menyebabkan diskontinuitas pada suatu segmen citra, atau pada suatu spatial window yang dievaluasi. Contoh *impulse noise* adalah *salt and pepper* seperti diilustrasikan pada Gambar 4 (a).
2. *Additive noise*, adalah sinyal-sinyal dengan magnitudo acak yang terdistribusi secara Gauss pada suatu citra digital. Contoh additive noise adalah derau putih (*white noise*) dan *Gaussian noise*, seperti diilustrasikan pada Gambar 4 (b).
3. *Multiplicative noise*, adalah suatu multiplikasi atau konvolusi dari beberapa noise dengan magnitudo, distribusi dan intensitas yang berbeda. Contoh speckle noise seperti ditunjukkan pada Gambar 4 (c).

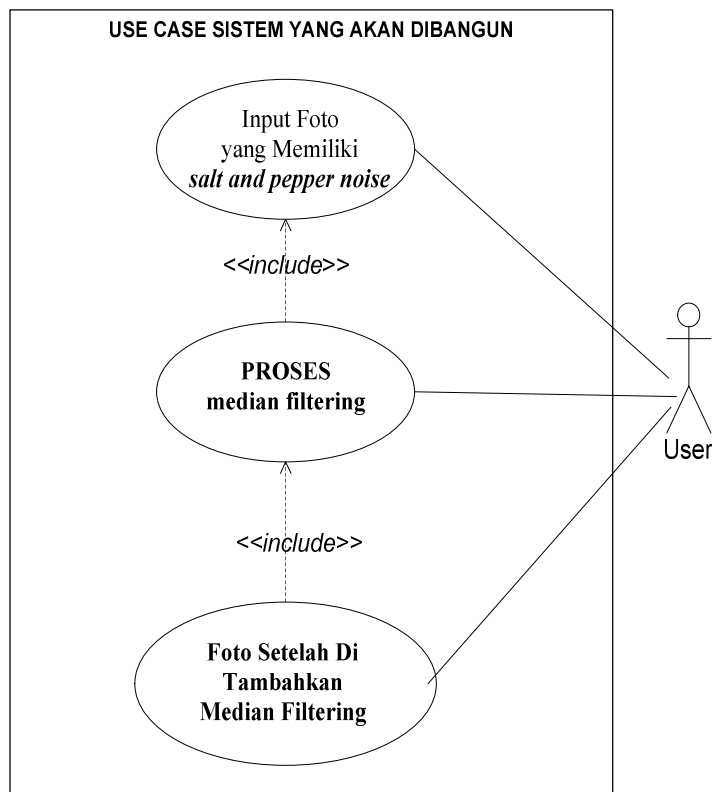


Gambar 4. Beberapa Jenis Noise Menurunkan Kualitas Citra Digital [1]

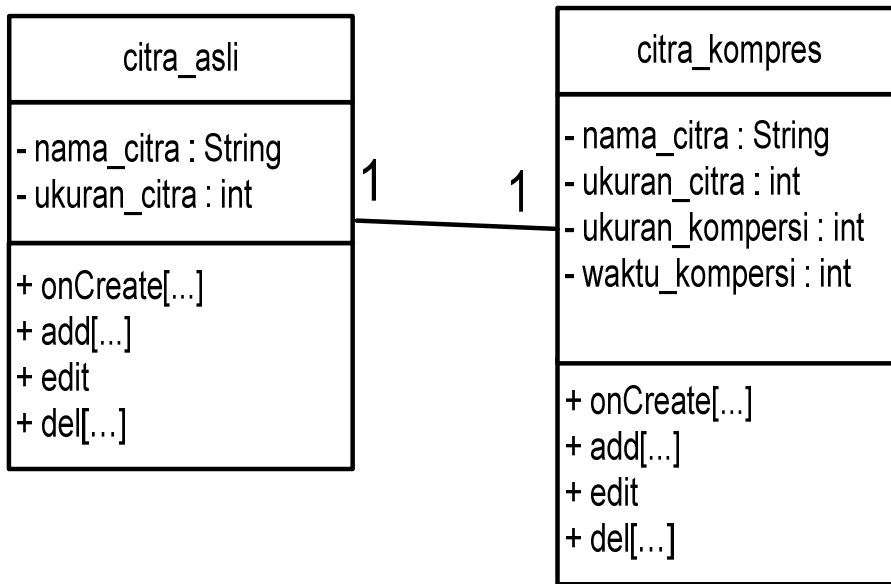
### 3. RANCANGAN DAN IMPLEMENTASI

#### 3.1 Desain Aplikasi

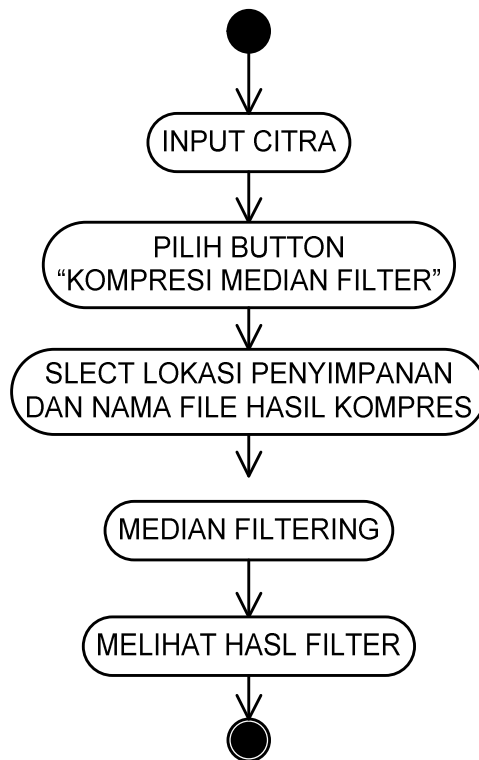
Desain rancangan umum aplikasi peningkatan kualitas citra menggunakan metode *median filtering* untuk menghilangkan *noise*, akan gambarkan dalam bentuk *use case*, *class diagram*, *activity* dan *Sequence Diagram* berikut ini:



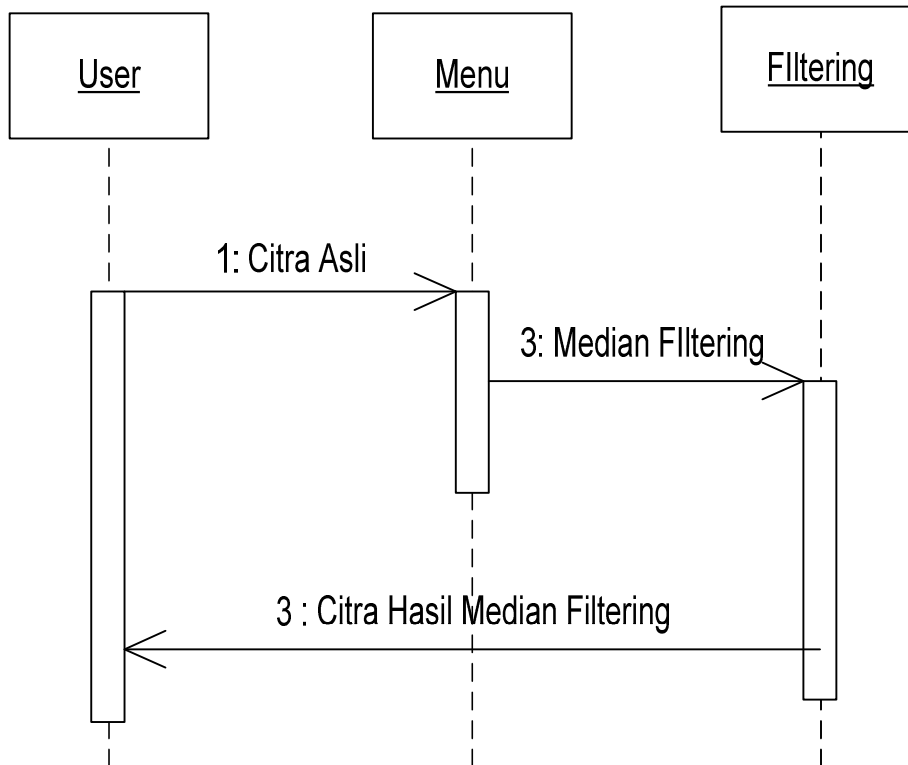
Gambar 5. Use Case Sistem Aplikasi



Gambar 6. Class Diagram Aplikasi



Gambar 7. Activity Diagram Aplikasi



Gambar 8. Sequence Diagram Aplikasi

**3.2 Hasil Implementasi Median Filtering**

Pada pembahasan ini penulis akan menjelaskan proses metode yang dilakukan median filtering dalam melakukan filtering pada objek citra awal sebelum difilter. Pada proses *median filtering* langkah-langkah algoritmanya adalah :<sup>[4]</sup>

1. Mencari nilai tengah dari pixel pixel pada window yang dievaluasi. Nilai ini selanjutnya menggantikan piksel pada titik tengah window sebagai koreksinya. Dari contoh yang sama dengan mean filter diatas, tahap 1 dibentuk window 3x3 dengan titik pusat di kordinat (1,1).

|   |   |     |   |     |     |   |   |
|---|---|-----|---|-----|-----|---|---|
| 2 | 2 | 2   | 3 | 4   | 5   | 2 | 3 |
| 2 | 3 | 5   | 2 | 144 | 2   | 5 | 2 |
| 3 | 3 | 3   | 3 | 4   | 5   | 6 | 7 |
| 3 | 4 | 240 | 3 | 2   | 3   | 5 | 6 |
| 2 | 3 | 4   | 6 | 8   | 222 | 2 | 5 |
| 2 | 4 | 6   | 7 | 9   | 3   | 5 | 7 |

$F(1,1) = \text{median}(2,2,2,2,3,3,3,3,5) = 3$  Median adalah nilai tengah (indeks ke-5) setelah diurutkan dari 9 data. Maka nilai koreksi untuk pixel pada posisi  $F(1,1) = 3$ , atau nilainya tidak berubah.

2. Selanjutnya titik pusat window bergeser ke posisi (1,2). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window.

|   |   |     |   |     |     |   |   |
|---|---|-----|---|-----|-----|---|---|
| 2 | 2 | 2   | 3 | 4   | 5   | 2 | 3 |
| 2 | 3 | 3   | 2 | 144 | 2   | 5 | 2 |
| 3 | 3 | 3   | 3 | 4   | 5   | 6 | 7 |
| 3 | 4 | 240 | 3 | 2   | 3   | 5 | 6 |
| 2 | 3 | 4   | 6 | 8   | 222 | 2 | 5 |
| 2 | 4 | 6   | 7 | 9   | 3   | 5 | 7 |

$F(1,2) = \text{median}(2,2,2,3,3,3,3,3,5) = 3$

Maka nilai koreksi pixel pada posisi  $F(1,2)$  adalah 3 dari nilai sebelumnya 5.

- Selanjutnya titik pusat window bergeser ke posisi (1,3). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window

|   |   |     |   |     |     |   |   |
|---|---|-----|---|-----|-----|---|---|
| 2 | 2 | 2   | 3 | 4   | 5   | 2 | 3 |
| 2 | 3 | 3   | 3 | 144 | 2   | 5 | 2 |
| 3 | 3 | 3   | 3 | 4   | 5   | 6 | 7 |
| 3 | 4 | 240 | 3 | 2   | 3   | 5 | 6 |
| 2 | 3 | 4   | 6 | 8   | 222 | 2 | 5 |
| 2 | 4 | 6   | 7 | 9   | 3   | 5 | 7 |

$F(1,3) = \text{median}(2,2,3,3,3,3,4,4,144) = 3$

Maka nilai koreksi pixel pada posisi  $F(1,3)$  adalah 3 dari sebelumnya bernilai 2.

- Kemudian titik pusat window bergeser ke posisi (1,4). Langkah yang sama dilakukan untuk menghitung nilai tengah dari titik pusat window.

|   |   |     |   |   |     |   |   |
|---|---|-----|---|---|-----|---|---|
| 2 | 2 | 2   | 3 | 4 | 5   | 2 | 3 |
| 2 | 3 | 3   | 3 | 4 | 2   | 5 | 2 |
| 3 | 3 | 3   | 3 | 4 | 5   | 6 | 7 |
| 3 | 4 | 240 | 3 | 2 | 3   | 5 | 6 |
| 2 | 3 | 4   | 6 | 8 | 222 | 2 | 5 |
| 2 | 4 | 6   | 7 | 9 | 3   | 5 | 7 |

$F(1,3) = \text{median}(2,3,3,3,4,4,5,5,144) = 4$

Maka nilai koreksi untuk pixel pada posisi  $F(1,4)$  adalah 4. Terlihat bahwa piksel yang tadinya berwarna mendekati abu-abu (nilai 144) dikoreksi menjadi sangat mendekati hitam (0) yaitu bernilai 4.

- Proses-proses diatas untuk berikutnya diulang terus hingga posisi piksel terakhir pada citra. Untuk citra ber-warna, nilai piksel yang dikoreksi dilakukan untuk setiap kanal warna, atau indeks dimensi matriks R,G,B

Berikut ini adalah proses implementasi pengujian aplikasi yang di bangun:

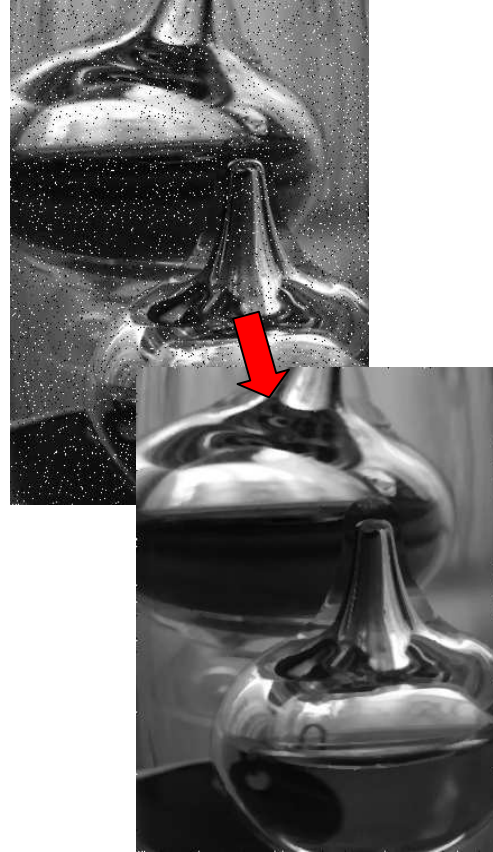
Tabel 1. Implementasi Pengujian Aplikasi

| Test Factor  | Hasil | Keterangan   |
|--|-------|--|
| Memasukkan citra asli, menekan tombol <b>Median Filter</b> dan melihat hasil filtering | ✓     | <b>Berhasil,</b><br>Karena setelah Aplikasi memproses gambar yang memiliki noise, maka gambar tersebut menjadi bersih dari noise |
| <i>Screen Shot</i>   |       |  |

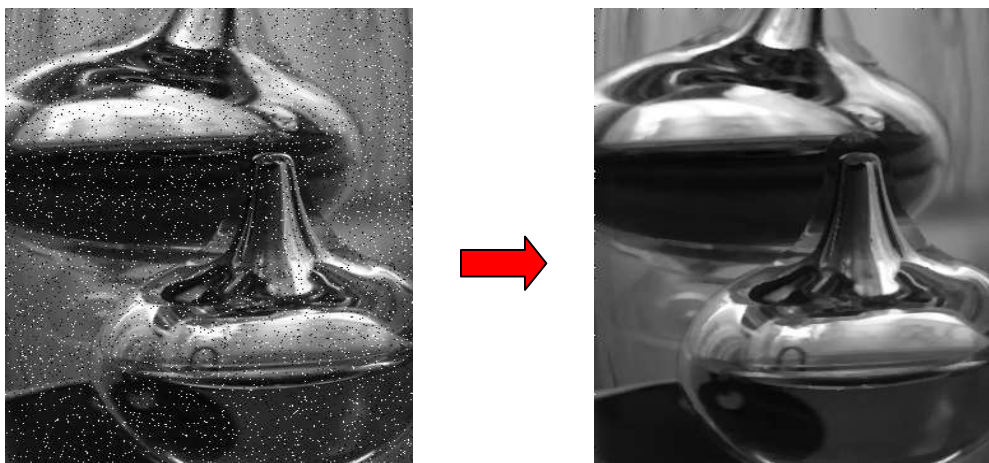
```

ant -f C:\\Users\\Lenovo\\Desktop\\ImageMedianSerial-master -Dnb.internal.action.name=run run
init:
Deleting: C:\\Users\\Lenovo\\Desktop\\ImageMedianSerial-master\\build\\built-jar.properties
deps-jar:
Updating property file: C:\\Users\\Lenovo\\Desktop\\ImageMedianSerial-master\\build\\built-jar.properties
Compiling 2 source files to C:\\Users\\Lenovo\\Desktop\\ImageMedianSerial-master\\build\\classes
compile:
run:
Durasi Gambar Memisahkan Serial 492
Durasi Proses Media Filtering Citra 224
Durasi Proses Media Filtering Citra 250
Durasi Proses Media Filtering Citra 189
Durasi Proses Media Filtering Citra 170
Durasi Proses Media Filtering Citra 170
Durasi Proses Media Filtering Citra 192
Durasi Proses Media Filtering Citra 170
Durasi Proses Media Filtering Citra 153
Durasi Proses Media Filtering Citra 150
Durasi Proses Media Filtering Citra 162
Durasi Proses Media Filtering Citra 149
Durasi Proses Media Filtering Citra 170
Durasi Proses Media Filtering Citra 250
Durasi Proses Media Filtering Citra 352
Durasi Proses Media Filtering Citra 297
Durasi Proses Media Filtering Citra 328
Durasi Proses Media Filtering Citra 150
Durasi Proses Media Filtering Citra 284
Durasi Proses Media Filtering Citra 162
Durasi Proses Media Filtering Citra 175
Durasi Proses Media Filtering Citra 222
Durasi Proses Media Filtering Citra 166
Durasi Proses Media Filtering Citra 246
Durasi Proses Media Filtering Citra 170
Durasi Proses Media Filtering Citra 168
Durasi Proses Media Filtering Citra 4634
PENGGABUNGAN CITRA.....
Durasi Gambar Menggabungkan Serial5771
Waktu Proses 11384
BUILD SUCCESSFUL (total time: 12 seconds)

```

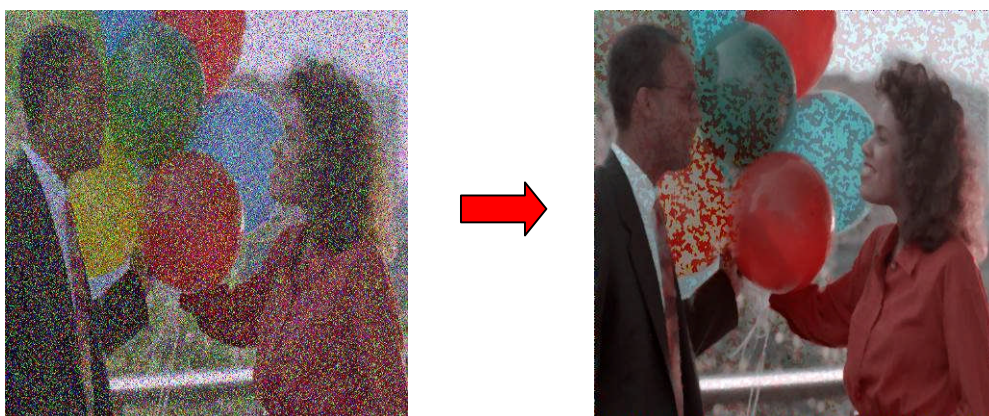


Setelah proses *Median Filter* selesai kita dapat melihat presentase perubahan gambar untuk kasus noise, presentase tersebut dapat berkurang hingga diatas 80% jika kita melihat secara kasat mata dari gambar awal (*noise*) dan gambar hasil setelah *median filter*. Sebagai contoh, kami akan memberikan hasil dari beberapa proses *median filtering* di bawah ini.

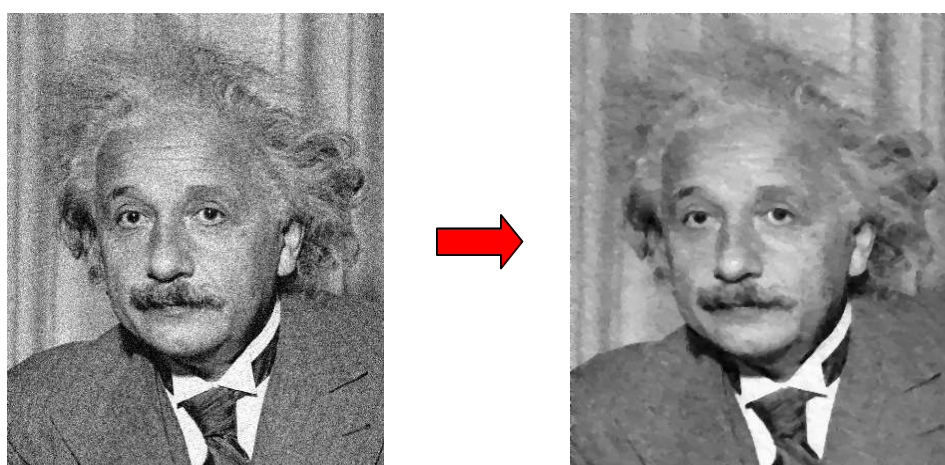


Gambar 9 Citra *Noise* dan setelah di *Median Filter* 1





Gambar 10 Citra *Noise* Berwarna dan setelah di *Median Filter 2*



Gambar 11. Citra *Noise* dan setelah di *Median Filter 3*

#### 4. KESIMPULAN

1. Aplikasi peningkatan kualitas citra ini dibangun dengan menggunakan alat perancangan UML dalam desain aplikasi dan menggunakan metode *median filtering* dalam proses peningkatan kualitas citra, dimana yang terjadi adalah terjadi pengurangan *noise* pada gambar.
2. Hasil yang didapatkan setelah pengujian aplikasi adalah *noise* dapat berkurang hingga diatas 80% jika gambar yang digunakan adalah gambar hitam putih, namun jika gambar yang digunakan memiliki warna hasil gaussian filter akan memberikan dampak pada warna citra tersebut

#### 5. SARAN

Untuk hasil yang optimal dapat digunakan citra biner atau *grayscale* sebagai media agar didapatkan hasil kompresi dan waktu yang lebih baik, karena pada citra tersebut tidak terdapat banyak warna sehingga memudahkan metode ini untuk bekerja.

**DAFTAR PUSTAKA**

- [1] Balza. A dan Firdausy. K, 2015, Teknik Pengolahan Citra Digital, Ardi Publishing, Yogyakarta.
- [2] Bambang Yuwono, 2015, *Image Smoothing Menggunakan Mean Filtering, Median Filtering, Modus Filtering dan Gaussian Filtering*, Telematika Vol. 7, No. 1, JULI 2010 : 65 – 75
- [3] Gonzalez, Rafael C., Richard E. Woods, 2015, *Digital Image Processing Third Edition*, Pearson Prentice Hall, Upper Saddle River, New Jersey. Cogito Smart Journal/VOL. 2/NO. 2/DESEMBER 2016
- [4] Ivan Maulana, Puluung Nurtantio, Andono, 2016, Analisa Perbandingan Adaptif Median Filter dan *Median Filter* Dalam Reduksi *Noise Salt & Pepper*,